The drone design concepts required two components, what services are required and how to deal with a mission plan. For the services, we had two options: have a single service that handles all instruction types, or many services that each handle a specific type of instruction. If we used a single service for all the instructions, we would only be required to have a single communication service running in xAPI, but we sacrifice both a minimal packet size and the modular design required to allow services to be built in the future. On the other hand, if we use many services that each handle an instruction, we have a larger amount of services running in xAPI, but reduce the complexity of each service and allow for new services to easily be created and added to the firmware.

With the mission plan design, with respect to communication, we had to decide how to send and store the entire mission plan. For storage we were directed to use non-volatile storage. As for the protocol for sending the mission plan, we could either send the entire mission plan to the drone at once or use an ACK/NACK system to verify that the drone receives each packet in order in addition to messages that indicate the start/end of a plan. Sending the entire mission plan without the use of an ACK/NACK would reduce the amount of time to send the plan to the drone, but we sacrifice the integrity of the plan. Using the ACK/NACK system, including start/end packets to indicate the start/end of a mission plan, allows for us to verify the mission plan has successfully been uploaded and maintained its integrity, but we sacrifice having a low amount of processing required and instead have a more complex mission plan installation process.

Given our requirements, our basic communication system design is that an instruction travels from the home computer over serial to the home Arduino, the serial service on the home Arduino then transfers the packet to the drone Arduino using the on-board xBee, and finally the drone Arduino receives the packet and the target service grabs the packet for processing. With this design, we use a set of services that each correspond to a specific drone function. For instance, if we send a takeoff packet to the drone, the takeoff service receives the packet and sends a takeoff command to the drone.

In addition to our basic drone communications, we also have designed a mission plan system that takes a sequence of instructions and stores them on the drone. We have yet to implement these services, but the design is as follows:

1. User creates a mission plan using the GUI
2. User initiates mission plan installation
3. A start mission plan install packet is sent over serial to the home Arduino
4. The packet is transferred over xBee to the drone Arduino
5. The drone stores the payload of the received packet, if packet is not start/end install
6. The drone Arduino sends an ACK to the home Arduino
7. The home Arduino sends an ACK over serial to the GUI
8. Repeat steps 3 – 6 with drone instruction packets followed by the end install packet
9. If an ACK is not received in the set timeout period, the last sent packet is re-sent

The communications system that needed to be implemented had to follow a set of given requirements. First, the physical hardware needed to include Arduino micro-controllers, xBee modules for the wireless communication using the ZigBee protocol, and a serial connection from the home computer and the home Arduino. We were also required to be running Brandon Ortiz’s xAPI, which at its roots is a simple service manager. These services are associated with two specific packet types used for local and external communication and are intended to be interpreted by the associated service. For instance, the Takeoff service packet is only processed by the Takeoff service. (Could be added somewhere if space allows)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* David Klingenberg

There is no budget for this project which limited components to parts on hand. The drone was left over from a previous project started by Brandon Ortize. All of the components used were salvaged from that project. Some sensors were purchased based on size, instrumentation integration, and price was a major factor. We had two design approaches. We could continue to use zip ties which were implemented on the drone we received at the beginning of the project or custom design mounting brackets to secure the equipment and instrumentation that were available.

The drone needed to be re-laid out in a more stable configuration so 3-D printed mounts were designed and affixed to the instrumentation, arduinos, and the drone's frame.